Midterm #1 Review

Be able to read and write code using:

1. Classes, construction, destruction
2. Class composition, initializer lists
3. Pointers and references
4. Dynamic memory allocation (new/delete)
5. Linked lists (doubly linked too)
6. Copy constructors and assignment operators

Review Problem #1

```
struct NODE
{
    int val;
    NODE *next;
};
```

Write a function called swapFirstTwo which swaps the first two items in a linked list (if there are at least two items).

```
class LinkedList
{
 public:
   ...
   void swapFirstTwo();
   ...
private:
    Node *m_head;
};
```

Problem #2

What does this program print?

```
class MagicPower
{
public:
    MagicPower() { cout << “MP\n”; }
    ~ MagicPower() { cout << “~MP\n”; }
};

class Wand
{
public:
    Wand() { cout << “Wand\n”; }
    ~Wand() { cout << “~Wand\n”; }
private:
    MagicPower myPower;
};

class Wizard
{
public:
     Wizard()  { cout << “Wizard\n”; }
     ~Wizard()  { cout << “~Wizard\n”; }

private:
     Wand myWand;
     Wand *secondWand;
};
```

```
int main()
{
     Wizard *carey;

     carey = new Wizard;

    Wizard david[2];

}
```

```
class Stomach
{
public:
    Stomach(string items[], int n)
     {
          count = n;

          if (n <= 3)  // if <= 3 items then use array
          {
            for (int i=0;i<n;i++)
                 contents[i] = items[i];
          }
          else          // otherwise use new to alloc
array
          {
               moreThanThree = new string[n];
               for (int i=0;i<n;i++)
                    moreThanThree[i] = items[i];
          }
     }

    ~Stomach()
    {
         if (count > 3)
              delete [] moreThanThree;
    }

private:

   string contents[3];
   string *moreThanThree;
    int count;
};
```

Problem #3

Show the copy constructor for the stomach class.

(At home, figure out the assignment operator)

```
class BadComedian
{
public:
    BadComedian(const string &joke, int times)
    {
        myJoke = joke;
        numTimes = times;
    }
    void tellJoke() const
    {
        for (int i=0;i<numTimes;i++)
            cout << myJoke << endl;
    }
private:
    string myJoke;
    int numTimes;
};
```

```
class NightClub
{
public:
    NightClub(int count)
    {
    }

    void doShow()
    {
        carey.tellJoke();
        david.tellJoke();
    }
private:
   BadComedian carey;
   BadComedian david;
};
```

Problem #4

Define a constructor for the NightClub class so it works as specified below:

Carey's joke should be:
"A man walks into a bar…ouch!"
and he should tell it count times.

David's joke should be:
"A fish swims into a wall…
Damn!" and he should tell it twice as many times as Carey.

Problem #1 Solution

```
void swapTwo()
{
    if (m_head == NULL || m_head->next == NULL)
        return; // not enough nodes, just return

    Node *temp;

   temp = m_head;            // get a temp pointer to the top node
   m_head = m_head->next;    // point our head pointer to the 2nd node
   temp->next = m_head->next;     // point our old head node to the 2nd node's
                             // child
   m_head->next = temp;
}
```

Problem #2 Solution

```
class MagicPower
{
public:
    MagicPower() { cout << “MP\n”; }
    ~ MagicPower() { cout << “~MP\n”; }
};

class Wand
{
public:
    Wand() { cout << “Wand\n”; }
    ~Wand() { cout << “~Wand\n”; }
private:
    MagicPower myPower;
};

class Wizard
{
public:
     Wizard()  { cout << “Wizard\n”; }
     ~Wizard()  { cout << “~Wizard\n”; }

private:
     Wand myWand;
     Wand *secondWand;
};
```

```
int main()
{
     Wizard *carey;

     carey = new Wizard;

    Wizard david[2];

}
```

```
MP
Wand
Wizard

MP
Wand
Wizard
MP
Wand
Wizard
~Wizard
~Wand
~MP
~Wizard
~Wand
~MP
```

Problem #3 Solution

```
class Stomach
{
public:

    Stomach(const Stomach &other)
     {
          count = other.count;
          if (count <= 3)
          {
            for (int i=0;i<count;i++)
                 contents[i] = other.contents[i];
          }
          else
          {
               moreThanThree = new string[count];
               for (int i=0;i<count;i++)
                    moreThanThree[i] =
other.moreThanThree[i];
          }
     }


private:

   string contents[3];
   string *moreThanThree;
    int count;
};
```

Problem #4 Solution

```
class BadComedian
{
public:
    BadComedian(const string &joke, int times)
    {
        myJoke = joke;
        numTimes = times;
    }
    void tellJoke() const
    {
        for (int i=0;i<numTimes;i++)
            cout << myJoke << endl;
    }
private:
    string myJoke;
    int numTimes;
};
```

```
class NightClub
{
public:
    NightClub(int count)
        : carey("A man walks into a bar...
ouch!",count),
        david("A fish swims into a wall...
Damn!",2*count)
    {
    }
    ...
private:
   BadComedian carey;
   BadComedian david;
};
```

Correct the NightClub class so it works properly:

Carey's joke should be:
"A man walks into a bar...ouch!"
and he should tell it count times.

David's joke should be:
"A fish swims into a wall...
Damn!" and he should tell it twice as many times as Carey.